

## GAMIFICAÇÃO DE ALGORITMOS

João Vittor Nalia; Luiz Carlos Begosso

*akiraseira@hotmail.com; lbegosso@femanet.com.br*

DOI: 10.5281/zenodo.16738940

**RESUMO:** A tecnologia da informação vem contribuindo para diversas áreas do conhecimento; em particular, o destaque é para o campo da educação. No que diz respeito ao ensino de programação de computadores, a literatura apresenta importantes esforços que visam auxiliar no processo de aprendizagem. Ensinar algoritmos e conceitos de programação para alunos do primeiro ano sempre foi um grande desafio para as universidades. Os novos estudantes de Ciência da Computação geralmente têm dificuldades em entender e abstrair as lógicas dos problemas. Uma alternativa que tem contribuído para o processo de ensino-aprendizagem é o uso de jogos, que contribuem para a mediação e o aprimoramento do processo de ensino-aprendizagem. Desenvolvemos um jogo chamado Maze para ajudar a ensinar conceitos que são considerados difíceis de entender pelos alunos dos cursos de Ciência da Computação. Este artigo apresenta a implementação do jogo Maze que objetiva ajudar os alunos em cursos de programação de computadores.

**PALAVRAS-CHAVE:** programação de computadores; ciência da computação; processo educacional; gamificação; greenfoot.

**ABSTRACT:** *Information technology has been contributing to various areas of knowledge; in particular, the field of education stands out. In what concerns the teaching of computer programming, literature contains important efforts that aim to assist in the learning process. Teaching algorithms and programming concepts for first year students has always been a great challenge for universities, new Computer Science students usually have difficulties in understanding and abstracting the problem logics. An alternative that has contributed to the teaching-learning process is the use of games, which contribute towards mediating and enhancing the teaching learning process. we have developed a game named Maze to help teach concepts that are considered difficult to understand by students of Computer Science courses. This paper presents the implementation of Maze game that aims to assist students in Computer Science courses.*

**KEYWORDS:** computer programming; computer science; educational process; gamification; greenfoot.

### 1. INTRODUÇÃO

A demanda pela formação de programadores tem aumentado nos últimos anos. O crescente de uso de computadores em todos os aspectos da vida humana e o uso da computação em eletrodomésticos, veículos, equipamentos médicos, máquinas, tratores e smartphones justificam esta demanda. Para a indústria de software a necessidade de novos funcionários supera a oferta produzida pelos variados cursos de computação.

Segundo SAARI, BLANCHFIELD e HOPKINS (2016) as maiores empresas desenvolvedoras de software têm pressionado governos em todo o mundo para elaborar planos estratégicos de forma a aumentar o número de programadores produzidos por seus sistemas educacionais.

Na academia, muito do trabalho para introduzir conceitos de programação se concentrou em como desenvolver a compreensão da programação em crianças e jovens sem experiência com tais conceitos. Nesta área destacam-se boas iniciativas que mesclam desde o desenvolvimento de jogos até a programação pura de código de alto nível. As experiências de BEGOSSO et al (2012) com as linguagens Python e Greenfoot, e BEGOSSO e SILVA (2013) com o Scratch ilustram bem este cenário.

BARATA et al (2013) ressaltam que os jogos possuem um grande potencial para melhorar a experiência de aprendizagem. Para estes autores, a gamificação tende a produzir melhorias no entendimento, no empenho e na motivação dos usuários quando são sujeitos num contexto de aprendizado.

Dentro deste contexto, o uso de jogos e ambientes de simulação tem ganhado espaço no ambiente acadêmico, especialmente para o ensino de conceitos de programação, de engenharia de

software e outras áreas correlatas. Esta metodologia é conhecida por “gamificação”. O conceito de gamificação está associado a utilização de elementos de jogos em contextos genéricos, podendo ser aplicado na área educacional ou em empresas. No contexto do presente projeto foram abordados três softwares para a produção de games, são eles: Greenfoot, Godot e Stencyl.

O Greenfoot é um software projetado para permitir que estudantes de cursos da área da computação possam obter experiência em programação orientada a objetos. O software permite o desenvolvimento de aplicações gráficas como criação de jogos em ambiente 2D. O Greenfoot foi desenvolvido em JAVA por pesquisadores da Universidade de Kent, e é direcionado para aquelas que pessoas que possuem alguma experiência com a programação de computadores (GREENFOOT, 2010).

Godot é um software, de código aberto, sob a licença do Instituto de Tecnologia de Massachusetts. Ele é destinado à criação de jogos 2D e 3D. Os jogos desenvolvidos no Godot podem ser exportados para as principais plataformas de desktop (Linux, MacOS, Windows), além de plataformas móveis (Android, iOS) e baseadas na Web (HTML5). De acordo com MANZUR e MARQUES (2018), Godot possui linguagem própria de programação denominada de GDScript que é semelhante a linguagem Python. Godot possui um motor gráfico que suporta características de transparências, reflexos, sombras dinâmicas, neblina e shaders. Para o desenvolvimento de jogos 2D e 3D, o Godot suporta características de sprites, animações, física e colisões. Sobre as animações, o Godot possui a capacidade de criar esqueletos (skeletons), inversão cinética (inverse kinematic) e deformação (morphing). Em relação à física, o Godot apresenta um sistema integrado para 2D e 3D que suporta colisões, objetos rígidos, cinética de corpos e simulação de viaturas.

De acordo com BORKWOOD (2013) o Stencyl é uma ferramenta para o desenvolvimento de jogos do tipo 2D que não requer conhecimento prévio de programação para utilizá-la. O mecanismo por trás do Stencyl é a combinação de blocos visuais para determinar o comportamento de personagens e objetos do jogo. A ferramenta está apta a gerar jogos para várias plataformas, como iOS, Android, Windows, Mac, Linux e Web.

Diante dos argumentos expostos e com a finalidade de explorar plataformas computacionais que proporcionem condições para o desenvolvimento de aplicações que facilitem e colaborem com o aprendizado de conceitos iniciais de programação de computadores é que surge o questionamento: a aplicação de games em disciplinas introdutórias de programação de computadores pode ajudar no entendimento e aprendizado dos conceitos iniciais e fundamentais da programação de computadores?

O primeiro passo para responder à pergunta formulada, foi eleger o software para implementar o jogo com características de gamificação para o ensino de algoritmos. Optou-se pelo Greenfoot pelo fato de existir uma comunidade atuante no suporte à ferramenta e por utilizar a linguagem Java para a escrita dos programas.

O presente trabalho tem por objetivo implementar um jogo com características de gamificação para o ensino de algoritmos para estudantes de séries iniciais de curso de computação.

Para atingir o objetivo estabelecido, o artigo foi dividido em 4 seções. A Seção 1, esta Introdução, estabelece os objetivos e a justificativa para a sua execução. A Seção 2, aborda os aspectos conceituais e práticos do ambiente Greenfoot, linguagem escolhida para desenvolver o game. O projeto e a implementação do game é apresentado na Seção

3. Finalmente, o trabalho alcança a Seção 4 com a apresentação das conclusões e os direcionamentos para futuras melhorias na aplicação.

## 2. GREENFOOT

O Greenfoot é um software projetado para permitir que alunos de cursos da área da computação possam obter experiência em programação orientada a objetos. O software permite o desenvolvimento de aplicações gráficas como criação de jogos em ambiente 2D. O Greenfoot foi desenvolvido em

JAVA por pesquisadores da Universidade de Kent, na Inglaterra, e da Universidade La Trobe, na Austrália, no ano de 2006.

A partir da visualização e interação de objetos, é possível criar minimundos e representar graficamente seus objetos.

O Greenfoot torna fácil a representação gráfica dos objetos, e controla a execução de uma vasta ordem de programas. Tudo isso pode ser visualizado em um plano bidimensional, facilitando a compreensão e a interação entre os objetos.

O software fornece um típico ambiente de desenvolvimento com a possibilidade de editar, compilar e executar o código-fonte criado pelo estudante. O que diferencia o Greenfoot de outros ambientes é a interação direta, por exemplo: ao instanciar um objeto, este pode ser colocado em qualquer lugar do “mundo”, e quando um método é invocado essa ação pode ser visualizada imediatamente.

O software pode ser usado tanto em escolas como em cursos avançados de universidades. O design e a facilidade de uso do software têm como objetivo chamar a atenção de alunos que ainda não estão comprometidos com programação e facilitar também o trabalho dos professores.

### 3. PROJETO E IMPLEMENTAÇÃO DO GAME

O projeto aqui desenvolvido trata da implementação do game denominado Maze. Para atender aos objetivos estabelecidos, o Maze foi construído com características de gamificação que auxilia estudantes de cursos da área de computação a aprenderem a programar. A Figura 1 ilustra as atuais funcionalidades do game:

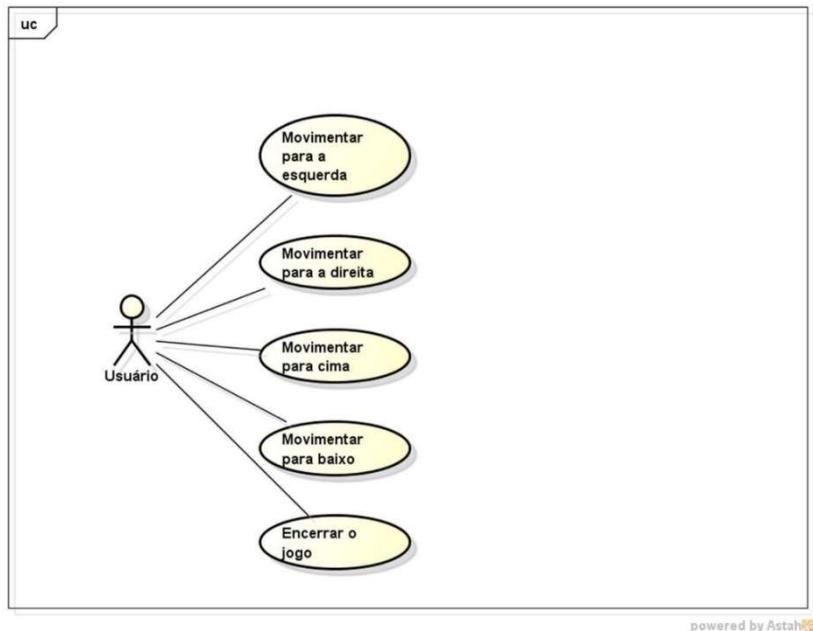


Figura 1 – Funcionalidades do Maze (Fonte: o autor)

No presente estágio de desenvolvimento do Maze foi possível implementar cinco funcionalidades capazes de executar o que está ilustrado na Figura 1. O aluno, ao interagir com o Maze, poderá compreender os conceitos de método e de objeto que são importantes aspectos da programação orientada a objetos. Isso significa dizer que os métodos são funções que realizam as ações de um objeto, ou seja, os métodos são responsáveis por definir os comportamentos do objeto.

A Figura 2 ilustra a página principal do Maze que é composta por um labirinto. O objetivo do jogador é fazer com que o personagem Rato evite o contato com o inimigo e consiga contabilizar

quatro Fatias de Queijo consumidas. Neste caso, o jogo é encerrado e o jogador recebe a informação de que concluiu a fase. De outro modo, se o jogador não comer as quatro fatias e encostar no inimigo o Maze informa que o jogador perdeu o desafio, o jogo é encerrado.

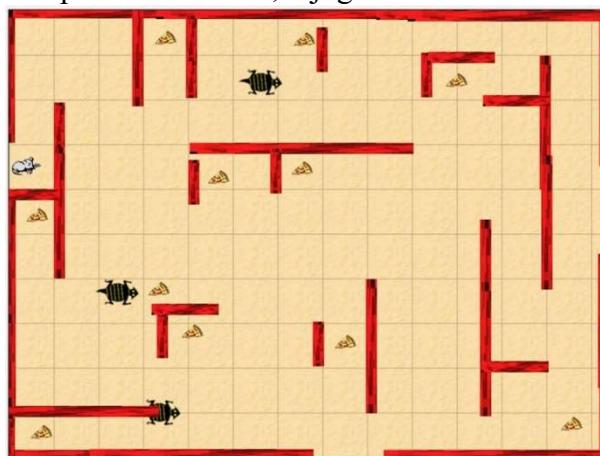


Figura 2 – Ambiente do Maze (Fonte: o autor)

A Figura 3 ilustra a forma pela qual o jogador interage com o Maze. Uma caixa de texto é disponibilizada ao jogador para que ele escreva os métodos que podem ser invocados pelo game. Conforme mencionado anteriormente, a atual versão do Maze possibilita que o jogador utilize cinco funcionalidades, são elas:

- hero\_moveleft(), desloca o personagem Rato uma casa para a esquerda;
- hero\_moveright(), desloca o personagem Rato uma casa para a direita;
- hero\_moveup(), movimenta o personagem para cima;
- hero\_movedown(), movimenta o personagem para baixo;
- stop(), encerra o jogo.

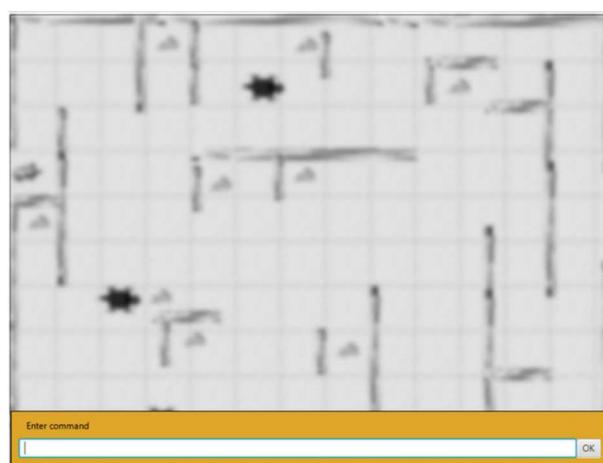


Figura 3 – Interagindo com o Maze (Fonte: o autor)

#### 4. CONCLUSÕES

Investir em novas tecnologias educacionais amplia oportunidades para estudantes na aquisição e compreensão de conteúdos abstratos e para os docentes a tecnologia pode tornar o processo de ensino-aprendizagem melhor.

O presente trabalho analisou buscou iniciativas na literatura para melhorar a experiência de aprendizagem de conteúdos voltados para a área de programação de computadores. Constatou-se que

a gamificação tende a produzir melhorias no entendimento, no empenho e na motivação dos estudantes quando são sujeitos num contexto de aprendizado. A pesquisa direcionou para a existência diversos softwares para a produção de games, são eles: Greenfoot, Godot e Stencyl. Neste contexto, optou-se pelo Greenfoot é um software projetado para permitir que estudantes de cursos da área da computação possam obter experiência em programação orientada a objetos.

Para atender aos objetivos estabelecidos desenvolveu-se um game denominado Maze, com características de gamificação, que auxilia os estudantes no aprendizado de programação de computadores. O jogo foi implementado como um labirinto de duas dimensões onde o jogador pode fazer o uso de cinco comandos para movimentar o personagem principal em todas as direções.

Acredita-se que o Maze tenha potencial para suportar o docente na melhoria do processo ensino-aprendizagem. Por outro lado, espera-se que o Maze motive os estudantes no aprendizado de novos conteúdos abstratos, como é o caso da programação de computadores.

Futuramente o Maze precisará receber novas funcionalidades de forma a ampliar os desafios do jogo. Para verificar a sua eficácia na metodologia de ensino e aprendizagem é de fundamental importância que este seja aplicado ao público alvo para também receber feedback de novos ajustes.

## REFERÊNCIAS

- BARATA Gabriel; GAMA, Sandra; JORGE, Joaquim; GONÇALVES, D. Engaging engineering students with gamification. In Games and virtual worlds for serious applications (vs-games), 2013 5th international conference, p. 1-8, Sept 2013.
- BEGOSSO, L. C., BEGOSSO, L. R., GONÇALVES, E. M. GONÇALVES, J. R. An approach for teaching algorithms and computer programming using Greenfoot and Python, 2012 Frontiers in Education Conference Proceedings, Seattle, WA, 2012, pp. 1-6.
- BEGOSSO, L. C. da SILVA, P. R. Teaching computer programming: A practical review. 2013 IEEE Frontiers in Education Conference (FIE), Oklahoma City, OK, 2013, pp. 508-510.
- BLIKSTEIN, P. O pensamento computacional e a reinvenção do computador na educação. 2008. Disponível em: <[http://www.blikstein.com/paulo/documents/online/ol\\_pensamento\\_computacional.htm](http://www.blikstein.com/paulo/documents/online/ol_pensamento_computacional.htm)>. Acesso em: 30 jun. 2018.
- BORKWOOD, L. Learning Stencyl 3.x Game Development: Beginner's Guide. Birmingham: Packt Publishing, 2013.
- GODOT. Godot Docs. Disponível em: <<http://docs.godotengine.org/en/3.0/about/introduction.html>>. Acesso em: 30 jun. 2018.
- KÖLLING, M. Introduction to Programming with Greenfoot: Object-Oriented Programming in Java with Games and Simulations. New Jersey: Pearson, 2010.
- MANZUR, A. MARQUES, G. Godot Engine Game Development in 24 Hours, Sams Teach Yourself: The Official Guide to Godot 3.0. Indianapolis: Pearson, 2018.
- SAARI, E. M., BLANCHFIELD P., HOPKINS, G. (2016) Computational Thinking: A Tool to Motivate Understanding in Elementary School Teachers. In: Zvacek S., Restivo M., Uhomoibhi J., Helfert M. (eds) Computer Supported Education. CSEDU 2015. Communications in Computer and Information Science, vol 583. Springer, Cham.